

REMARKS/ARGUMENTS

Overview of the Office Action

Claims 1 and 17 were rejected under 35 U.S.C. § 102(b) as being anticipated by Juster (U.S. Patent No. 5,724,406).

Claims 1, 2, 8-15, 17, 18, and 20-24 were rejected under 35 U.S.C. § 102(b) as being anticipated by Satter et al. (U.S. Patent No. 5,243,643).

Claims 3-5 and 19 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Satter in view of Matthews et al. (U.S. Patent No. 4,652,700).

Claims 6 and 7 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Satter in view of Matthews and further in view of Weber (U.S. Patent No. 6,094,239).

Claims 16 and 25 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Satter in view of Chencinski et al. (U.S. Patent No. 5,355,406).

Status of the Claims/Amendments

Claims 1-25 are pending.

Claims Rejected Under 35 U.S.C. § 102(b)

Regarding Claims 1 and 17

Claims 1 and 17 were rejected under 35 U.S.C. § 102(b) as being anticipated by Juster (U.S. Patent No. 5,724,406).

Remarks on the Rejection

In regard to Claim 1, the Examiner concludes that "Juster teaches using various call processing primitives (CPP) for customizing call process service (column 5, lines 12-32)" and

that “Juster’s software [module] includes functions of call flows (column 5, lines 26-29), codes (a software comprises computer executable codes), and a list of names (names of variables, functions, users, etc.) and a modifiable list of corresponding DTMF signal identifier (column 5, lines 23-26).” The Examiner also reaches a similar conclusion in regard to Claim 17.

However, the Applicants respectfully submit that Juster does not teach or suggest “a table with a list of names and a modifiable list of corresponding DTMF signal identifiers” as set forth in Claims 1 and 17. This modifiable list of corresponding DTMF signal identifiers, referred to in the specification as a “Customization List,” comprises “a table with a list of names recognized by the development system and a modifiable list of corresponding DTMF signals identifiers” that enable a “customer” (a non-programmer user of the claimed functionality) to “change the mapping between caller-entered DTMF signal and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers in the Customization List(s)” (Specification, page 14, lines 24-29). Thus, in the invention of the present application, the DTMF signal identifiers are mapped to specific modules, each of which performs specific desired functions, and remapping DTMF signals to the modules requires nothing more than modifying this table (the Customization List).

The invention of Juster, on the other hand, comprises a plurality of modules (referred to as “CPPs”) that “perform one single identifiable operation, such as recording a message, playing a prompt, collecting a digit, *reading DTMF sequences*, etc.” (col. 5, lines 27-29) (emphasis added). Moreover, for example, “a user develops a voice mail messaging application having the necessary voice prompts and DTMF responses *by selecting appropriate CPPs [modules] that generate those prompts and tone responses*” (col. 5, lines 23-26) (emphasis added). In other

words, each CPP in the invention of Juster has hard-coded DTMF signal identifiers, and thus it would be necessary to select an alternate CPP having the same functionality but different DTMF signals when a user desires to change which DTMF signals initiate the desired CPP functionality. Therefore, in order to achieve the same level of flexibility afforded by a single module with DTMF signal mapping in a Customization List as disclosed in the present application, the invention of Juster would necessarily require a library of CPPs, one for each possible combination and/or permutation of possible DTMF signal identifiers that a customer may find desirable to use to call the specific functionality.

Nowhere does Juster suggest that a user can actually modify a single CPP such that specific DTMF signals corresponding to specific functionality for that specific CPP are thereby changed. On the contrary, Juster instead teaches away from a DTMF-modifiable CPP and instead suggests that a user must select a specific (or “appropriate”) CPP to achieve specific functionality, including hard-coded DTMF mapping, which is predefined for each individual CPP. Consequently, the inability of a user to change the specific DTMF signals corresponding to specific functionality for a specific CPP in the invention of Juster requires a programmer must to first develop, for utilization by a user, a library of CPPs for each possible combination of DTMF signals mapped to specific CPP functionality from which the user can select the DTMF mapping desired. The only other alternative is custom development of CPPs on an as-needed basis where “such diverse requirements are handled by modifying the application code as required for each different incarnation...[t]his solution, however, is unsatisfactory since it creates a complex maintenance dilemma...[t]he changes may be simple in nature but the maintenance of

a set of changes to the standard product for every customer can nonetheless become unduly burdensome and expensive” (Specification, page 4, lines 13-21).

In contrast, Claim 1 of the present invention claims “a module comprising call flows, code and a customization list; wherein the customization list comprises a table with a list of names and a modifiable list of corresponding DTMF signal identifiers, whereby *the particular customer is permitted to change the mapping between caller-entered DTMF signals and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers*” (emphasis added). For example, the user might have favorite numbers or combinations of numbers pertaining to specific functionality (for example, pressing “*6” to delete a voicemail message), or that the conversion from letters to numbers on the telephone keypad could provide for mnemonic enhancement for the user's benefit (for example, pressing “335”—the numbers corresponding to the letters “DEL”—to delete a voicemail message). The benefits afforded by the present invention are not available via the invention of Juster.

This specific limitation, which enables a user to modify the specific DTMF signals corresponding to specific functionality for that specific module, precludes the need for a library of modules with various DTMF signal mapping, and thereby clearly differentiates the present invention from the invention of Juster. Moreover, Claim 17 includes a similar limitation, and is likewise distinguishable from the teachings of Juster.

Remarks on Examiner's “Response to Arguments”

To further clarify the patentably-distinct differences between Juster and the invention of the current Application, a point-by-point analysis of the Examiner's key comments in the

“Response to Arguments” section of the Office Action may be beneficial. (For convenience, the Examiners comments from the Office Action are italicized.)

- *“The applicant argues that Juster does not disclose a table linking a DTMF identifier and its corresponding action, and DTMF identifiers are hard-coded into CPPs (functions or modules), thus requires a library of CPPs, one of each possible combination/permutation of possible DTMF signal identifiers with various functions, and making Juster's system lacking flexibility.”* (Office Action, page 9, lines 15-19.) Although essentially correct, the Applicants do not broadly contend that Juster’s system lacks flexibility in the absolute, but only does so relative to the invention of the present Application. More accurately, it is the Applicant’s contention that the system of Juster does indeed provide *some* flexibility, but only to the extent that multiple CPPs of identical functionality for different DTMF responses are compiled and provided to a user. For example, if the Juster invention has a CPP for Functionality A mapped to DTMF “1”, another CPP for Functionality A mapped to DTMF “2”, and a third CPP for Functionality A mapped to DTMF “3”, then the user in Juster would have three choices for which DTMF signal would be used to call Function A. However, if the system of Juster does not have yet another CPP for Functionality A mapped to DTMF “4”, then the user could not select—and therefore would not be able to use—DTMF “4” to call Function A. This is because the DTMF signal for a specific CPP is hard-coded in that CPP and cannot be changed by a user because a “user,” as that term is used in Juster, is equivalent to a “customer” of the present Application—that is, a person who would utilize the invention but who is not a programmer and who would not compile modules or CPPs. In contrast, the invention of the present Application enables a customer to map (and re-map) specific DTMF signal identifiers to specific modules (each of which performs specific desired functions) where remapping DTMF signals to the modules requires nothing more than modifying a Customization List table. Thus, in the present Application, Function A needs only be embodied in a single module, but that single module can then be associated with any specific DTMF signals the customer desires.

- *“However, Juster teaches: ‘a user develops a voice mail messaging application having the necessary voice prompts and DTMF responses by selecting appropriated [sic] CPPs that generate those prompts and tone responses’ (column 5, lines 23-26) (also quoted by the applicant in the Remark, page 3).” (Office Action, page 9, lines 20 to page 10, line 2.)*
Applicants have quoted this section of the Juster reference both for what it says and what it does not say. More specifically, a user must select an appropriate CPPs to generate the prompts and tone responses—not “change” or “modify” or “remap” the DTMF signals for a single CPP. Selecting an appropriate CPP necessarily means that there are a plurality from which to choose, and thus a selection inherently implies that no modifications to any CPP are occurring when “a user develops a voice mail messaging application” (id.). Therefore, the Examiner’s reliance on this section to sustain the position that Juster teaches a means for modifying CPPs is unfounded.
- *“Juster clearly teaches that the voice mail messaging system outputs voice prompts and responses (action taken by the messaging system) in responds [sic] to a DTMF identifier entered by a user, and by inherency, it is a table linking DTMF identifiers to their corresponding actions (defined by CPPs) as Juster states: ‘each CPP performs one simple identifiable operation, such as recording a message, playing a prompt, collecting a digit, reading DTMF sequence etc’ (column 5, lines 27-29).” (Office Action, page 10, lines 2-8.)*
While Juster may in fact teach a voice mail messaging system that outputs voice prompts and responses in response to a DTMF identifier entered by a user, the “inherent” table linking DTMF identifiers to their actions is a function-implementation table—e.g., DTMF “1” to listen to messages, DTMF “2” to send a message, DTMF “3” to change answering options, etc.—and the user of Juster builds this table by selecting the CPPs with the desired functionality and having the desired DTMF signals and forms these CPPs into a linked list (and thus the reason why the table is “logical” and not actual, since such a table arguably exists for the invention of present Application as well, and in addition to the table of the Customization List). If the user later wants to use a different DTMF signal to call specific functionality, the user must replace the existing CPP in the linked list with another—e.g., replace CPP-A1 (Functionality A, DTMF “1”) with CPP-A5 (Functionality A, DTMF “5”) in

the system. Conversely, the table of Juster is not a actual DTMF mapping table as used by the invention of the present Application—e.g., mapping DTMF “1” to module M, DTMF “2” to module N, etc.—and it is this DTMF mapping that enables a customer to change the DTMF signal for a specific module without having to replace that module—e.g., mapping DTMF “3” to module M, DTMF “1” to module O, etc.

- *“Juster clearly teaches a friendly environment in that a user, without programming knowledge, may customize a link list (a table, see page 8, lines 17-24 of applicant’s specification) of DTMF identifiers and corresponding actions by selecting a desired CPP (action) corresponds [sic] to a particular DTMF identifier (column 4, lines 49-54; column 5, lines 23-26, 45-53).”* (Office Action, page 10, lines 8-12.) In this regard, Applicants once again point out that “changes” to the function-based table in Juster require the removal of old CPPs and the addition of new CPPs in the linked list, whereas in the invention of the present Application the modules remain the same but only the DTMF data in the Customization List is changed.

Based on the foregoing, Applicants respectfully submit that only by impermissibly viewing the Applicant’s disclosure in hindsight could a person of skill in the art read into the Juster reference the functionality and benefits described and disclosed in the present Application, and that absent the present Application the Juster reference in no way teaches a system where DTMF signals are mapped to modules in a Customization List.

Request for Claims to Immediately Issue

Applicants respectfully request that this rejection under § 102(b) be withdrawn and that Claims 1 and 17 be allowed to issue.

Regarding Claims 1, 2, 8-15, 17, 18, and 20-24

Claims 1, 2, 8-15, 17, 18, and 20-24 were rejected under 35 U.S.C. § 102(b) as being anticipated by Satter et al. (U.S. Patent No. 5,243,643).

In regard to Claim 1, the Examiner concludes that “Satter discloses a voice processing system with configurable caller interfaces, comprising: a module [caller interface] comprising call flow functions, code and customization list (column 28, lines 18-37); wherein the customization list comprises a table of names and modifiable list of corresponding DTMF signal identifiers, where a customer is permitted to change the mapping between caller entered DTMF signal, and the corresponding actions taken by a voice messaging system (column 28, lines 1-3, 18-51; column 29-30, Vector Pogrecln).” The Examiner also reaches a similar conclusion in regard to Claim 17.

However, the Applicants respectfully submit that Sattar does not teach or suggest “a table with a list of names and a modifiable list of corresponding DTMF signal identifiers” as set forth in Claims 1 and 17. This modifiable list of corresponding DTMF signal identifiers, referred to in the specification as a “Customization List,” comprises “a table with a list of names recognized by the development system and a modifiable list of corresponding DTMF signals identifiers” that enable a user to “change the mapping between caller-entered DTMF signal and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers in the Customization List(s)” (Specification, page 14, lines 24-29). Thus, in the invention of the present application, the DTMF signal identifiers are mapped to specific modules, each of which performs specific desired functions, and remapping DTMF signals to the modules requires nothing more than modifying this table (the Customization List).

The invention of Sattar, on the other hand, takes an entirely different approach where the DTMF signal mapping is stored in compiled software that can only be changed by an application developer. Specifically, DTMF signal mapping is stored in the software listing of Appendix A

of Sattar (see col. 28, lines 18-24). Appendix A is (a) a listing of vectors (Sattar's equivalent to modules of the present application) used to control caller interfaces to the voicemail system (see col. 28, lines 6-10). Vectors are stored in an Application State Logic Table (AST) which is compiled rather than used in source code form (see col. 11, line 66 through col. 12, line 10) and that are generated by an application editor (APE) (see col. 14, lines 67 through col. 15, line 19). Significantly, vectors are only modifiable and adaptable in the C-language (col. 11, lines 42-49; col. 12, lines 21-26), and thus changing vectors requires recompilation. More specifically, to change DTMF signals, a "user"—which is actually an application developer (see col. 28, lines 1-6)—edits the vectors using APE (col. 28, lines 16-24) which, again, is an editor for source code that must be compiled for use.

Nowhere does Sattar suggest that a non-programmer user can actually modify (aside from rewriting the C-language code and recompiling) a single vector such that specific DTMF signals corresponding to specific functionality for that specific vector are thereby changed. On the contrary, Sattar instead teaches away from a DTMF-modifiable vector and speaks only to an application developer hard-coding DTMF mapping in the C-programming language for compilation (thus resulting in a predefined vector from the end-user perspective). Consequently, the inability of a user to change the specific DTMF signals corresponding to specific functionality for a specific vector in the invention of Sattar requires a programmer to first develop, for utilization by an end-user, a library of vectors for each possible combination of DTMF signals mapped to specific vector functionality from which, presumably, the user can select the DTMF mapping desired. The only other alternative is custom development of vectors on an as-needed basis where "such diverse requirements are handled by modifying the

application code as required for each different incarnation...[t]his solution, however, is unsatisfactory since it creates a complex maintenance dilemma...[t]he changes may be simple in nature but the maintenance of a set of changes to the standard product for every customer can nonetheless become unduly burdensome and expensive” (Specification, page 4, lines 13-21).

In contrast, Claim 1 of the present invention claims “a module comprising call flows, code and a customization list; wherein the customization list comprises a table with a list of names and a modifiable list of corresponding DTMF signal identifiers, whereby *the particular customer is permitted to change the mapping between caller-entered DTMF signals and the corresponding actions taken by the messaging system by modifying the list of DTMF signal identifiers*” (emphasis added). To recite the sample example previously used herein, the user might have favorite numbers or combinations of numbers pertaining to specific functionality (for example, pressing “*6” to delete a voicemail message), or that the conversion from letters to numbers on the telephone keypad could provide for mnemonic enhancement for the user's benefit (for example, pressing “335”—the numbers corresponding to the letters “DEL”—to delete a voicemail message). The benefits afforded by the present invention are not available via the invention of Sattar.

This specific limitation, which enables a user to modify the specific DTMF signals corresponding to specific functionality for that specific module, precludes the need for a library of modules with various DTMF signal mapping, and thereby clearly differentiates the present invention from the invention of Sattar. Moreover, Claim 17 includes a similar limitation, and is likewise distinguishable from the teachings of Sattar.

In addition, and for the reasons discussed herein above pertaining to the Juster reference, Applicants respectfully submit that only by impermissibly viewing the Applicant's disclosure in hindsight could a person of skill in the art read into the Sattar reference the functionality and benefits described and disclosed in the present Application, and that absent the present Application the Sattar reference in no way teaches a system where DTMF signals are mapped to modules in a Customization List.

For these reasons, Applicants respectfully request that, in regard to Claims 1 and 17, the rejection under § 102(b) be withdrawn. Moreover, given that Claims 2, 8-15, 18, and 20-24 are claims that depend from Claim 1 or Claim 17, Applicants further request that the rejection under § 102(b) be withdrawn. Finally, in light of the traversal of the forgoing rejection, Applicants humbly submit that Claims 1, 2, 8-15, 17, 18, and 20-24 should be allowed to issue.

Claims Rejected Under 35 U.S.C. § 103(a)

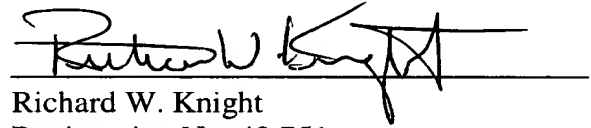
Claims 3-7, 16, 19, and 25 have been rejected as follows: Claims 3-5 and 19 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Satter in view of Matthews et al. (U.S. Patent No. 4,652,700); Claims 6 and 7 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Satter in view of Matthews and further in view of Weber (U.S. Patent No. 6,094,239); and Claims 16 and 25 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Satter in view of Chencinski et al. (U.S. Patent No. 5,355,406). However, Claims 3-7, 16, 19, and 25 depend from independent Claim 1 or Claim 17 and, as such, are allowable as dependent claims depending from an allowable claim. Therefore, Applicants respectfully request that the rejections against these claims under 35 U.S.C. § 103(a) be withdrawn and that these claims be allowed to issue.

CONCLUSION

Based on the reasons and rationale set forth herein, Applicants respectfully submit that the objections and rejections have been overcome and, accordingly, Applicants request that the objections and rejections be withdrawn and that the claims be allowed to issue. Should the Examiner have any questions, comments, or suggestions that would expedite the prosecution of the present case to allowance, Applicants' undersigned representative earnestly requests a telephone conference at (206) 332-1394.

Respectfully submitted,

Date: April 5, 2004


Richard W. Knight
Registration No. 42,751

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

Correspondence Address:
Unisys Corporation
Attn: Office of the General Counsel (Patents)
Unisys Way, MS/E8-114
Blue Bell, PA 19424-0001